

## I) Wiederholung: Einführung in die OOP mit Java

1. Definieren und vergleichen Sie die beiden Entwurfsprinzipien der Software-Erstellung:
  - *Prozedurales bzw. imperatives Programmieren*
  - *Objektorientiertes Programmieren*
2. Welches sind die Vorteile des objektorientierten Programmierstils? Begründung!
3. Definieren Sie die beiden Begriffe *Klasse* und *Objekt*.
4. Erläutern Sie die Begriffe: (Objekt-)Referenz und *Heap*. Visualisieren Sie diesen Sachverhalt am Beispiel der Objektdекlaration von *k1* in *MainClass* (vgl. Programmcode).
5. Was versteht man unter *Datenkapselung*? Welchen Vorteil bringt dieses Konzept?
6. Wodurch unterscheidet sich ein *Konstruktor* von normalen Methoden?

7. Gegeben sei folgendes Programmgerüst:

```
public class Konto
{
    private String kNr;
    private String kInhaber;
    private double kStand;
    public Konto()
    { kStand=0; }
    public Konto(String kInhaber, String kNr, double kStand)
    { this.kInhaber=kInhaber;
      this.kNr=kNr;
      this.kStand=kStand; }
    public void einzahlen(double betrag)
    { kStand=kStand+betrag; }
    public void auszahlen(double betrag)
    { kStand=kStand-betrag; }
    public double getKStand()
    { return kStand; }
}

public class MainClass {
    public static void main(String[] args)
    { Konto k1 = new Konto();
      Konto k2 = new Konto("Huber", "12022200", 5);
      k1.setKNr("40223307");
      k1.einzahlen(10000);
      System.out.println("Kontostand k1: "+k1.getKStand());
      k1.auszahlen(100);
      System.out.println("Kontostand k1: "+k1.getKStand());
      System.out.println("Kontostand k2: "+k2.getKStand());
    }
}
```

- Kennzeichnen Sie die Eigenschaften und Methoden der Klasse *Konto*!
- *Konto* hat 2 Konstruktoren: Woran erkennt die VM den passenden Konstruktor?
- Welches Problem entsteht, wenn der parameterleere Konstruktor gelöscht wird?
- Welche Bedeutung hat das Schlüsselwort *this*?
- Codieren Sie alle notwendigen Getter- und Setter-Methoden für die Klasse *Konto* und erstellen Sie eine weitere (überladene) Methode *setKNr(long kNr)*.

## II) Weitere Übungsaufgaben

Beim objektorientierten Programmieren wird vor allem die Modellierung des Programms in den Mittelpunkt gestellt und erscheint zunächst aufwändiger als beim prozeduralen Programmieren. Selbstverständlich müssen aber auch hier (wie früher auch) Algorithmen codiert werden.

### Aufgabe 1

Das Geldinstitut will einen Überblick haben, wie viele Konten insgesamt vorhanden sind. Hierfür soll ein Zähler implementiert werden, der bei jeder Kontoerstellung inkrementiert wird. Dieser Zähler soll durch eine entsprechende Methode *getCounter()* ausgelesen werden können.

### Aufgabe 2

Die Konten sollen nun in einem Array verwaltet werden. Ändern Sie das Java-Programm entsprechend ab. Außerdem soll jetzt zusätzlich noch verhindert werden, dass die gleiche Kontonummer doppelt vergeben wird. Wenn eine Kontonummer bereits existiert, dann soll wieder die Standardnummer gesetzt werden.

### Aufgabe 3

Alle Konten mit Ihren Eigenschaften sollen auf Bildschirm ausgegeben werden.

### Aufgabe 4

Entwickeln Sie für die Klasse *Konto* eine weitere Methode *public String berechne IBAN(String kNr, String BLZ)* zur Erzeugung und Rückgabe der zugehörigen IBAN.

Den Algorithmus zur IBAN-Berechnung können Sie unter <https://www.sparkonto.org/manuelles-berechnen-der-iban-pruefziffer-sepa/> nachlesen.

Überprüfen Sie die korrekte Funktion anhand eines der zahlreichen vorhandenen Tools im Internet.